CRYPTOGRAPHIC RANDOMNESS TESTING PADA ALGORITMA *BLOCK CIPHER* CAMELLIA MENGGUNAKAN UJI *COVERAGE*

Adrian Admi[1)]

[1)]*Lembaga Sandi Negara*
Ragunan, Jakarta Selatan
[1)]admi.adrian@gmail.com

Abstract— **Telah dilakukan pengujian *coverage* terhadap algoritma *block cipher* Camellia. Pengujian *coverage* bertujuan untuk melihat korelasi antara *input* dan *output* algoritma Camellia dengan kondisi tertentu. Dalam penelitian ini, suatu algoritma *block cipher* dinyatakan lulus uji *coverage* jika nilai *coverage* yang muncul memenuhi uji kesesuaian *chi square* dan memenuhi *p-value* yang telah ditetapkan untuk ukuran sampel 2^{20}. Algoritma Camellia dinyatakan lulus uji *coverage* setelah didapatkan hasil pengujian dengan nilai uji kesesuaian *chi square* sebesar 2.140701 (Camellia-128), 5.696789 (Camellia-192), 5.943172 (Camellia-256) dengan *p-value* 0.709899 (Camellia-128), 0.222965 (Camellia-192), 0.203433 (Camellia-256).**

Keywords— **Uji *Coverage*, *Block Cipher*, Camellia**

I. PENDAHULUAN

Salah satu layanan dari kriptografi adalah keamanan informasi (*confidentiality*). Salah satu *tools* yang dapat digunakan untuk mengamankan informasi adalah algoritma *block cipher*. Algoritma *block cipher* bekerja dengan merubah informasi yang sebelumnya dapat dibaca (*plaintext*) menjadi informasi yang tidak dapat dibaca (*ciphertext*), proses ini disebut enkripsi. Lalu proses pengembalian informasi dari tidak terbaca (*ciphertext*) menjadi dapat terbaca (*plaintext*) disebut proses dekripsi.

Perkembangan algoritma *block cipher* yang cukup pesat menyebabkan desain *block cipher* banyak dikembangkan oleh peneliti dan instansi pemerintah maupun swasta. Salah satunya adalah algoritma Camellia yang mengadopsi struktur *Feistel* yang dikembangkan oleh *Nippon Telegraph and Telephone Corporation* (NTT) dan Mitsubishi *Electric Corporation*.

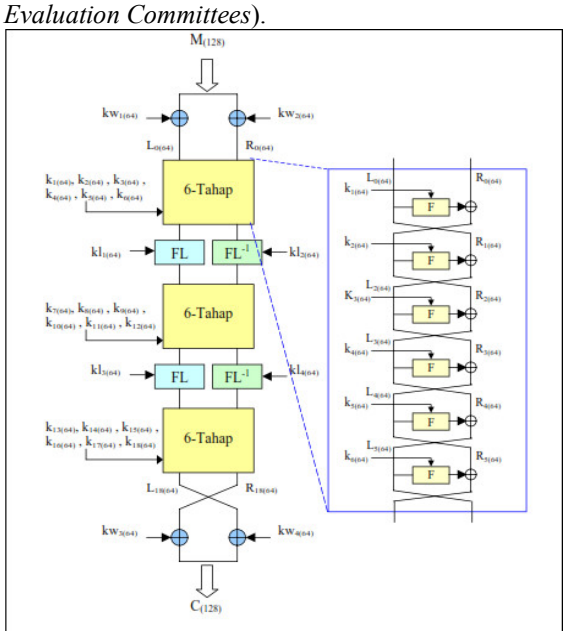
Sejalan dengan perkembangan desain *block cipher* yang makin beragam, metode evaluasi dan pengujian *block cipher* juga berkembang sesuai desain yang ada. Salah satu pengujian *block cipher* yang dikembangkan oleh Fatih Sulak dalam Tesisnya adalah *Cryptographic Randomness Testing* untuk algoritma *block cipher*.

Salah satu *Cryptographic Randomness Testing* yang dapat diterapkan pada algoritma *block cipher* adalah uji *coverage*. Dalam penelitian ini, uji *coverage* dilakukan terhadap algoritma Camellia untuk melihat nilai *coverage* dari algoritma tersebut. Uji tersebut melihat apakah algoritma Camellia yang merupakan algoritma *standard*NESSIE dan CRYPTREC memiliki salah satu sifat dari *Cryptographic Randomness*.

II. ALGORITMA CAMELLIA

Camellia adalah sebuah algoritma kriptografi *block cipher* yang memiliki ukuran blok 128-bit dengan panjang kunci 128-bit, 192-bit, atau 256-bit. Camellia pertama kali dikembangkan secara bersama oleh NTT dan Mitsubishi *Electric Corporation* pada tahun 2000.

Camellia sudah banyak diteliti oleh beberapa ahli di bidang kriptografi. Algoritma ini sudah dipilih sebagai rekomendasi algoritma kriptografi primitif oleh NESSIE (*New European* Schemes *for Signatures, Integrity and Encryption*) dan juga termasuk salah satu algoritma yang diterapkan pada sistem *e-Government* di Jepang yang dipilih oleh CRYPTREC (*Cryptography Research and Evaluation Committees*).



Gambar 1. Enkripsi Camellia dengan kunci 128 bit

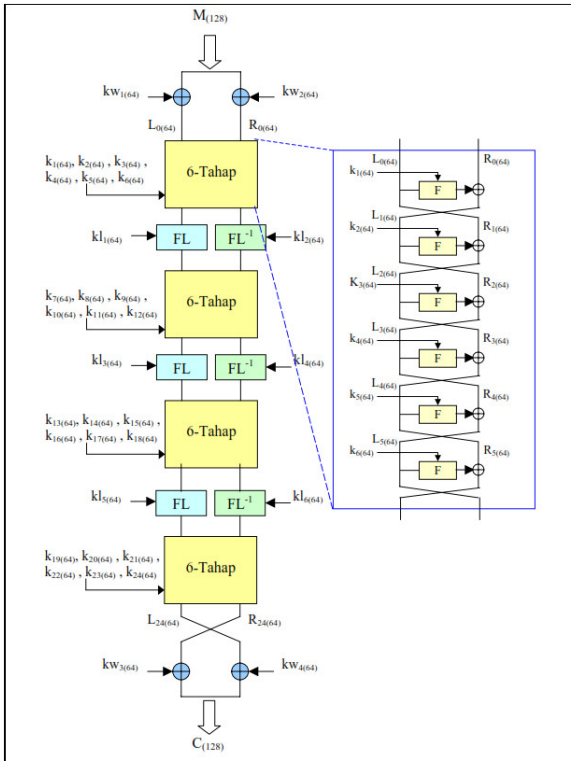
1. Enkripsi Camellia 128 Bit

Gambar 1. menunjukkan prosedur enkripsi untuk kunci 128 bit. Bagian pengacakan data memiliki struktur 18 tahap (*round*) *feistel*

dengan 2 lapisan (layer) fungsi FL/FL^{-1} setelah *round* ke-6 dan *round* ke-12, dan operasi XOR 128 bit sebelum tahap pertama dan setelah tahap terakhir. Bagian penjadwalan kunci membangkitkan subkunci $k_{wt(64)}(t=1, 2, 3, 4)$, $k_{u(64)}(u=1, 2, ..., 18)$, dan $k_{lv(64)}(v=1, 2, 3, 4)$ dari kunci rahasia K. Secara lebih jelasnya dapat dilihat pada Gambar 1.

2. Enkripsi Dengan Kunci 192 Bit Dan 256 Bit

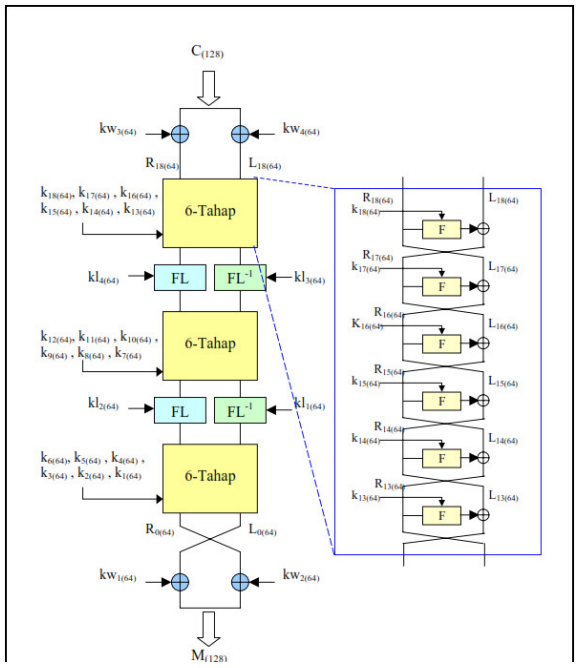
Bagian pengacakan *input* memiliki struktur 24 *round* feistel dengan 3 buah lapisan fungsi FL/FL^{-1}setelah *round* ke-6, ke-12, dan ke-18, dan operasi XOR 128-bit sebelum *round* pertama dan setelah *round* terakhir. Bagian penjadwalan kunci membangkitkan subkunci $kw_{t(64)}(t=1, 2, 3, 4)$, $k_{u(64)}(u=1, 2, ..., 24)$, dan $kl_{v(64)}(v=1, 2, ..., 6)$ dari kunci rahasia K. Secara lebih jelasnya dapat dilihat pada Gambar 2.



Gambar 2. Enkripsi Camellia kunci 192 dan 256 bit

3. Prosedur Dekripsi Camellia 128 Bit

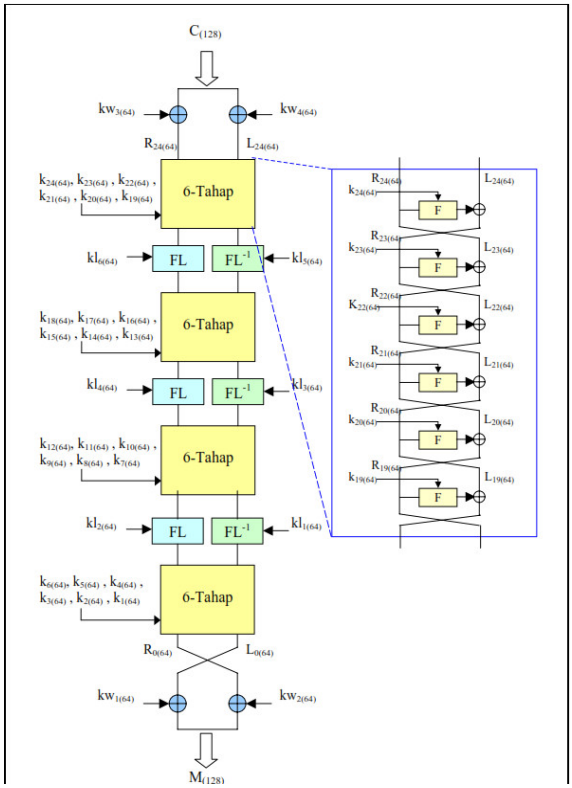
Prosedur dekripsi dari Camellia dilakukan dengan jalan yang sama dengan prosedur enkripsi dengan membalik urutan dari subkunci. Gambar 3. menunjukkan prosedur dekripsi untuk kunci 128-bit. Bagian pengacakan data memiliki struktur 18 *round* feistel dengan 2 lapisan (layer) fungsi FL/FL^{-1} setelah tahap ke-6 dan tahap ke-12, dan operasi XOR 128-bit sebelum *round* pertama dan setelah *round* terakhir. Bagian penjadwalan kunci membangkitkan subkunci $k_{wt(64)}(t=1, 2, 3, 4)$, $k_{u(64)}(u=1, 2, ..., 18)$, dan $k_{lv(64)}(v=1, 2, 3, 4)$ dari kunci rahasia K. Lihat Gambar 3.



Gambar 3. Dekripsi Camellia 128 bit

4. Prosedur Dekripsi Camellia 192 Dan 256 Bit

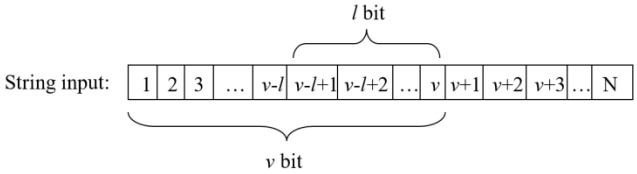
Bagian pengacakan data memiliki struktur 24 *round* feistel dengan 3 lapisan (layer) fungsi FL/FL^{-1} setelah tahap ke-6, ke-12 dan tahap ke-18, dan operasi XOR 128-bit sebelum *round* pertama dan setelah *round* terakhir. Bagian penjadwalan kunci membangkitkan subkunci $k_{wt(64)}(t=1, 2, 3, 4)$, $k_{u(64)}(u=1, 2, ..., 24)$, dan $k_{lv(64)}(v=1, 2, ..., 6)$ dari kunci rahasia K. Lihat Gambar 4.



Gambar 4. Dekripsi Camellia 192 dan 256 bit.

III. METODE UJI COVERAGE

Pada uji *coverage,* untuk setiap string *input* acak, dimodifikasi dengan semua kemungkinan kombinasi yang dilakukan pada *l* bit yang dipilih dari *v* bit pertama dari string *input* dengan ukuran *N*. Dalam hal ini *l* bit yang akan dimodikasi disebut sebagai bit-bit aktif dan *v-l* bit sisanya adalah bit *in*-aktif.



Dengan demikian, dari 1 string input akan diperoleh 2^l*input* dengan cara memodifikasi *l* bit meggunakan semua kemungkinan kombinasinya. Selanjutnya pada setiap *input* baru hasil modifikasi tersebut dimasukkan ke fungsi *f* (dalam hal ini *block cipher*) yang memetakan *input* menjadi *output*. Lalu akan diperoleh 2^l*output*, yaitu $Z^{(1)}, Z^{(2)}, Z^{(3)}, \ldots, Z^{(2^l)}$. Berdasarkan semua *output* tersebut, selanjutnya dicek jumlah *output* yang berbeda pada *l* bit yang telah ditentukan sebelumnya. Banyaknya *output* yang berbeda inilah yang disebut sebagai *coverage*. Dengan demikian, nilai *coverage*, *K*, yang mungkin adalah 1, 2, 3, …, 2^l.

Pada *coverage test*, percobaan tersebut diulang sebanyak 2^{20} dengan mengambil rangkaian bit yang berbeda, sehingga dihasilkan sejumlah nilai *coverage*. Selanjutnya nilai-nilai *coverage* tersebut ditabulasikan dengan rentang kelas dan peluang setiap rentang adalah sebagai berikut:

TABEL 1. RENTANG DAN PROBABILITAS UJI *COVERAGE* UNTUK SAMPEL 2^{20}

	l = 12 bit pertama	
No	Range *Coverage*	Probabilitas
1	1-2572	0.199176
2	2573-2584	0.204681
3	2585-2594	0.197862
4	2595-2606	0.203232
5	2607-4096	0.195049

Frekuensi setiap kelas dihitung sesuai dengan banyaknya nilai yang masuk ke dalam kelas tersebut, sehingga akan diperoleh f_1, f_2, f_3, f_4 dan f_5 serta total frekuensi $F=f_1+f_2+f_3+f_4+f_5$. Berdasar nilai *F* dan peluang setiap kelas (yang dihitung dengan formula iteraktif di atas), maka dihasilkan frekuensi teoritis, yaitu e_1, e_2, e_3, e_4 *dan* e_5. Setelah diperoleh frekuensi empiris *f*, dan teoritis *e*, maka dilanjutkan dapat dihitung uji kesesuaiannya dengan *chi square*.

1. Algoritma Uji
   Sebanyak *R* kali percobaan, lakukan langkah-langkah dibawah ini:
   a. Bangkitkan rangkaian acak sepanjang *N*. Nilai *N* adalah panjang *inputblock cipher*.
   b. Lakukan perubahan pada 12 bit pertama dengan menggunakan semua kemungknan kombinasi nilai 12 digit biner, sedangkan bit lainnya tetap. Oleh karena itu diperoleh 2^{12} sekuen *input* biner dengan panjang *N*.
   c. Hitung hasil enkripsi dari 2^{12}*input* tersebut menggunakan *block cipher* yang akan diuji.
   d. Hitung *coverage*: banyaknya 12 bit awal yang berbeda dari total 2^{12}*output* yg ada.
   e. Lakukan pendataan nilai *coverage* yang didapat, dan kelompokkan ke dalam interval sesuai dengan Tabel 1.
   f. Langkah a s.d. e dilakukan sebanyak R (2^{20}) kali hingga didapat frekuensi dari masing-masing interval, dengan total frekuensi sebanyak 2^{20}.
   g. Lakukan uji kesesuaian dengan *Chi square.*
   h. Tetapkan taraf nyata, $\alpha = 0{,}01$, pada derajat bebas: *k-1, k* adalah banyaknya kelas. Hitung nilai *Chi square* hitung:
   $$\chi^2_{hitung} = \sum_{i=1}^{5} \frac{(f_i - e_i)^2}{e_i}$$
   i. Bandingkan nilai *Chi square* tabel dengan *Chi square* hitung, dan hitung *p-value* dari nilai *Chi square* hitung tersebut. Jika:
   $$\chi^2_{hitung} < \chi^2_{Tabel(k-1,\alpha)} \text{ atau}$$
   $$P_{value(k-1,\alpha)} > \alpha$$
   maka algoritma*block cipher* tersebut dinyatakan lulus uji *coverage*.

IV. HASIL PENGUJIAN DAN ANALISIS UJI

1. Hasil Pengujian *Coverage* Camellia
   Uji *coverage* pada algoritma Camellia dilakukan terhadap tiga panjang kunci Camellia, yaitu 128, 192, dan 256 bit. Jumlah *plaintext* yang digunakan untuk uji *coverage* sebanyak 2^{20} (1048576) yang dibangkitkan secara acak. Menurut Fatih Sulak, suatu algoritma *block cipher* dinyatakan lulus uji *coverage* jika memiliki nilai *p-value* di atas 0,01.

TABEL 2. OBSERVED FREQUENCY COVERAGE CAMELLIA 128 BIT

No	Range *Coverage*	*Expected Freq*	*Observed Freq*
1	1-2572	208851	208466
2	2573-2584	214623	215132
3	2585-2594	207473	207268
4	2595-2606	213104	213124
5	2607-4096	204523	204586

   a. Camellia Kunci 128 bit
      Melalui sampel sebanyak 1048576, didapatkan sebanyak 1048576 nilai *coverage*, lalu dicek nilai-nilai*coverage*tersebut dan dikelompokkan

ke dalam lima rentang sesuai Tabel 1. Untuk hasil observasi pada Camellia 128 bit, tertera pada Tabel 2.

Berdasarkan hasil observasi tersebut, dapat dihitung nilai*Chi Square*sebagai berikut:

$$\chi^2_{hitung} = \frac{(208466-208851)^2}{208851} - \frac{(215132-214623)^2}{214623} - \frac{(207268-207473)^2}{207473} - \frac{(213124-213104)^2}{213104} - \frac{(204586-204523)^2}{204523} = \mathbf{2.140701}$$

Melaluisimulasi program didapatkan *p-value*dari nilai *Chi Square* tersebut yaitu0.709899 (>0.01), dengan demikian Camellia 128 dikatakan lulus uji *coverage*.

b. Camellia Kunci 192 bit
Hasil observasi *coverage* Camellia 192 bit tertera pada Tabel 3.

TABEL 3. OBSERVED FREQUENCY COVERAGE CAMELLIA 192 BIT

No	Range *Coverage*	*Expected Freq*	*Observed Freq*
1	1-2572	208851	209090
2	2573-2584	214623	214949
3	2585-2594	207473	206987
4	2595-2606	213104	212437
5	2607-4096	204523	205113

Berdasarkan hasil observasi tersebut, dapat dihitung nilai*Chi Square*sebagai berikut:

$$\chi^2_{hitung} = \frac{(209090-208851)^2}{208851} - \frac{(214949-214623)^2}{214623} - \frac{(206987-207473)^2}{207473} - \frac{(212437-213104)^2}{213104} - \frac{(205113-204523)^2}{204523} = \mathbf{5.696789}$$

*P-value*dari nilai *Chi Square* tersebut adalah 0.222965 (>0.01), dengan demikian Camellia 192 dikatakan lulus uji *coverage*.

TABEL 4. OBSERVED FREQUENCY COVERAGE CAMELLIA 256 BIT

No	Range *Coverage*	*Expected Freq*	*Observed Freq*
1	1-2572	208851	208349
2	2573-2584	214623	214301
3	2585-2594	207473	208339
4	2595-2606	213104	213341
5	2607-4096	204523	204246

c. Camellia Kunci 256 bit
Hasil observasi *coverage* Camellia 256 bit tertera pada Tabel 4.

Berdasarkan hasil observasi tersebut, dapat dihitung nilai*Chi Square*sebagai berikut:

$$\chi^2_{hitung} = \frac{(208349-208851)^2}{208851} - \frac{(214301-214623)^2}{214623} - \frac{(208339-207473)^2}{207473} - \frac{(213341-213104)^2}{213104} - \frac{(204246-204523)^2}{204523} = \mathbf{5.943172}$$

*P-value*dari nilai *Chi Square* tersebut adalah 0.203433 (>0.01), dengan demikian Camellia 256 dikatakan lulus uji *coverage*.

2. Analisis Pengujian *Coverage* Algoritma Camellia
   a. Nilai *p-value* dari uji *coverage* Camellia 128,192 dan 256 bit seluruhnya di atasα(0,01) sehingga sebaran *coverage* nya memenuhi interval distribusi normal yang ditentukan. Hasil observasi frekuensi nilai *coverage* yang didapat disesuaikan dengan frekuensi harapan menggunakan uji kesesuaian *Chi Square*, dan didapatkan hasil *observed frequency* nya masih sesuai.
   b. *Coverage* yang bagus menandakan bahwa *ciphertext* yang dihasilkan oleh Algoritma Camellia 128, 192, dan 256 bit akan memunculkan setidaknya 63% dari 12 bit pertama dari seluruh kemungkinan 12 bit. Nilai tersebut merupakan nilai *coverage* yang baik sesuai dengan pernyataan Fatih Sulak dalam tesisnya.
   c. Waktu eksekusi pengujian *coverage* dengan *plaintext* sebanyak 2^{20} sekitar 3000 detik (sekitar 50 menit). Waktu tersebut masih cukup rasional dan *feasible* untuk dilakukan dengan sarana komputasi saat ini.

V. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa, algoritma Camellia lulus *Cryptographic Randomness Testing* untuk uji *coverage*. Hasil uji *coverage* untuk panjang kunci 128, 192, dan 256 bit dinyatakan lulus dengan memenuhi syarat nilai *p-value* lebih besar dari 0.01.

Untuk tahapan selanjutnya, masih ada beberapa uji *Cryptographic Randomness* yang dapat diterapkan pada algoritma Camellia, yaitu uji SAC, *linear span*, dan *collision*. Pengujian-pengujian tersebut dapat dilakukan pada penelitian selanjutnya untuk mengetahui lebih jauh tingkat

Cryptographic Randomness dari algoritma *Block Cipher* Camellia.

DAFTAR PUSTAKA

Aoki, Kazumaro, dkk. *Specification of Camellia – a 128-bit Block Cipher*. Nippon Telegraph and Telephone Corporation, Mitsubishi Electric Corporation. Jepang. 2000-2001.

Federal Information Processing Standard and Technologies (FIPS). *Announcing the Advanced Encryption Standard* (AES). National Institute of Standard and Technology (NIST). USA. 2001.

Menezes, Alfred J., Paul C. Van Oorschot, Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC press LLC: Boca Raton. 1997.

Sulak, Fatih. *Statistical Analysis of Block Cipher and Hash Function*. Thesis of Doctoral Program at The Graduate School of Applied Mathematics of Middle East Technical University. 2011.